Secure Database Commitments and Universal Arguments of Quasi Knowledge

Melissa Chase (Microsoft Research)
Ivan Visconti (University of Salerno)

Size hiding protocols

- Traditional secure computation:
  - Parties learn nothing more than f(x,y)
  - *And the size of the inputs*

All existing *definitions* and *constructions* reveal input size

- Sometimes the size of the inputs may be private/confidential
  - No fly list, phishing lists, company databases

- Can we hide the size of the players inputs and still achieve strong security?

Yes! We will show a construction for secure commitments which hides the input size

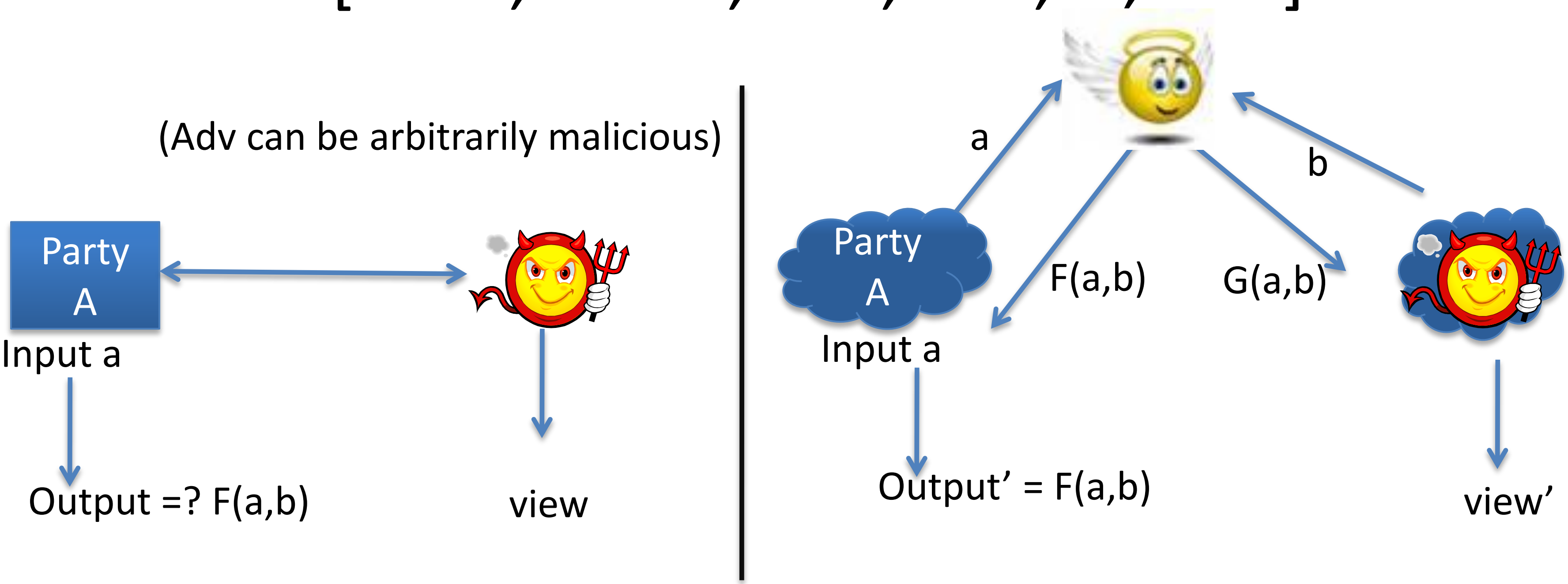
Size hiding protocols

- Input size must be poly(k), but can be any polynomial
  - No polynomial upper bound
- Prior work
  - Zero Knowledge Sets (ZKS) [MRK03, …]
  - Size-hiding Set Intersection [ADT11]
  - Branching Programs [IP07]

Semi honest security and/or ad hoc definitions

- How do we usually define secure computation?
  - Real/Ideal model

Real/Ideal model
[GL90, MR91, B91, C95, …, G04]



- Idea: Any attack in the real world could also occur in the ideal world

Traditionally: All parties know the size of the inputs (part of the description of F)

Our work

- Goal: realize size-hiding secure computation
  - Real/ideal model with malicious adversaries
- We focus on a very basic functionality: *Commitments*
- We give
  - Real/ideal model definition for size hiding (database) commitments
  - Constant-round construction based on CRHFs
  - Key building block: Universal Argument of Quasi Knowledge

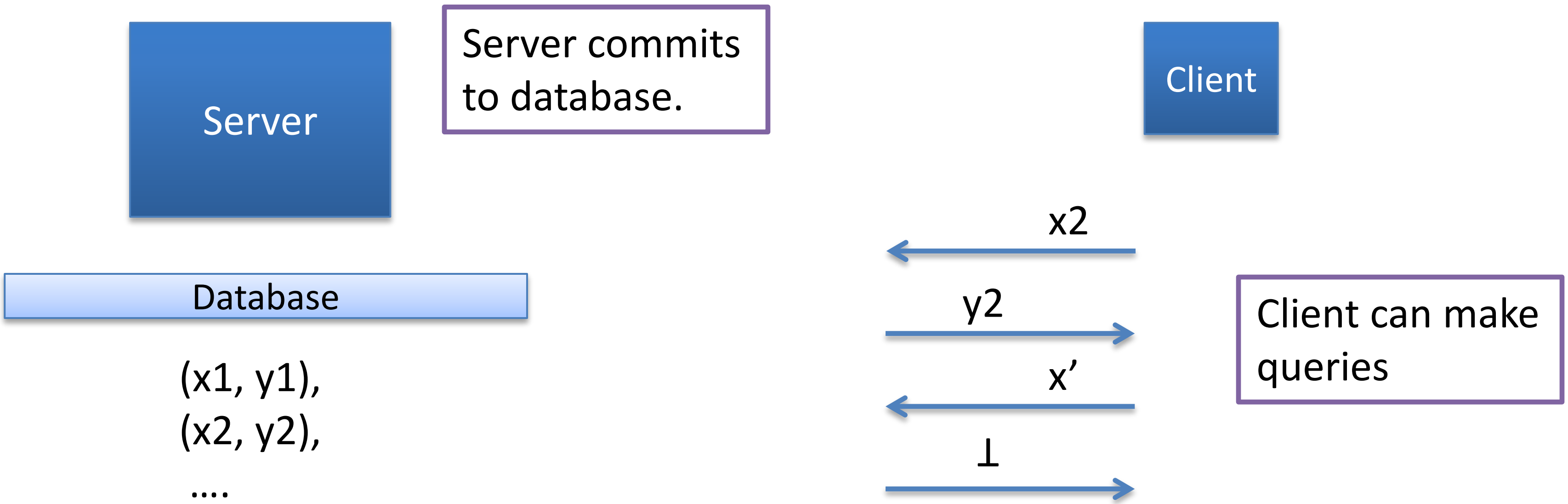
First size-hiding protocols in the real/ideal model

Roadmap

- Defining database commitments in the real/ideal model
- Universal Arguments of Knowledge [BG02]
  - and why they don't directly apply
- A new tool: Universal Arguments of Quasi Knowledge
- Constructing secure database commitments

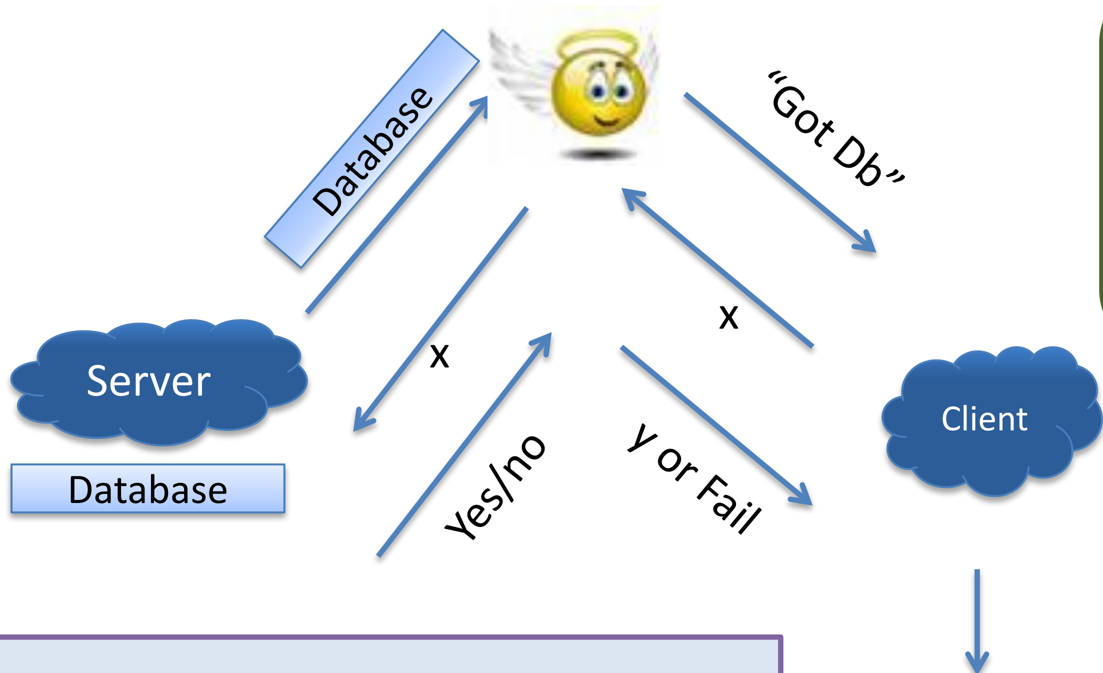
Secure Database Commitments

- High level idea: server can
  - Commit to a large input
  - Open it incrementally
- Elementary databases as in MRK03:



- Server can't change his mind later - must answer consistently with original database
- Client only learns answers to his queries (Does not learn size of database!)

Secure Database Commitments: Ideal model (first attempt)



y, …

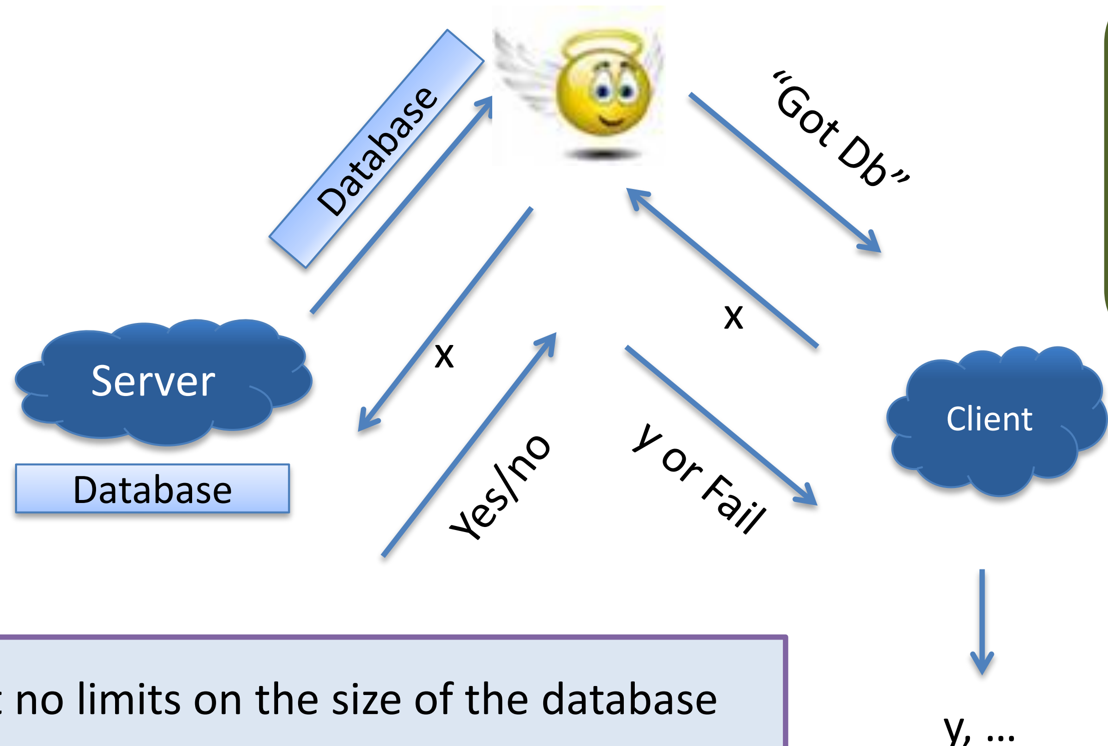
Could be viewed as ideal/real world version of Zero Knowledge Sets [MRK03]

We put no limits on the size of the database

y can be
$\perp$ = "not in Db"

- Server must "know" what he's committing to from the beginning
- Client only learns answers to queries
- Query responses must be consistent with original database

Secure Database Commitments: Ideal model (first attempt)



Could be viewed as ideal/real world version of Zero Knowledge Sets [MRK03]

We put no limits on the size of the database

y can be
⊥ = "not in Db"

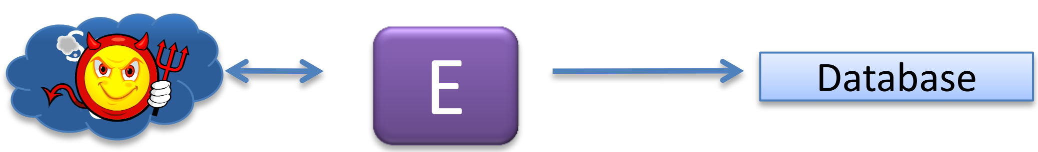
- **Server must "know" what he's committing to from the beginning**
- Client only learns answers to queries
- Query responses must be consistent with original database

Implications of the definition

Server must "know" what he's committing to from the beginning

What happens when we want to realize this part of the definition?

Standard approach: there exists an extractor



Traditionally: commit + proof of knowledge, encryption, etc

But, communication needs to be independent of input size!

Recall: can't assume a fixed poly(k) upperbound

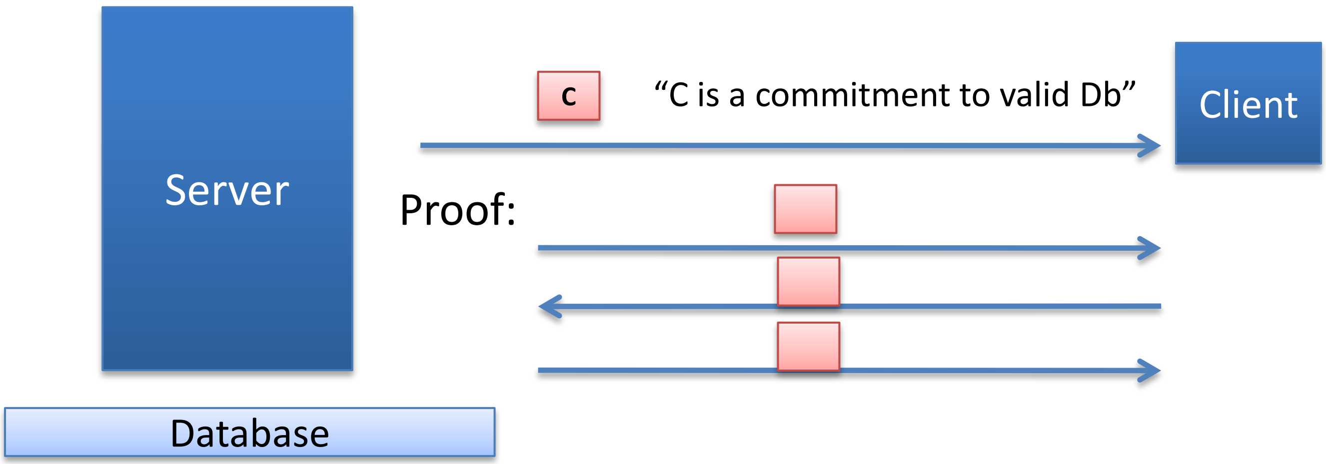
Implications of the definition

- We need a proof system where
  - 1) communication is much shorter than the witness
  - 2) must be a proof of knowledge so we can extract the witness
- Then perhaps we can apply commit and prove methodology
- Is there such a proof system?
  - What about Universal Arguments of Knowledge [BG02]?

Universal Arguments [BG02]

Original Application: Concurrent ZK

- Short proofs (even when witness is long)



- Witness Indistinguishability
  - Can't tell which database was used for proof

Database1 OR Database2

- Proofs of Knowledge weak

UAoK: Weak proof of knowledge

Why is it weak?

- E produces a circuit describing the witness

where w_i is the i-th bit of the witness

Address with modification to functionality

- If A produces a good proof with probability 1/p, E produces a good circuit with probability 1/p'
- We can't tell when C is a good circuit
  - (extracting t bits may take too long)
- E needs to be given a lower bound on the success probability of A
  - (running time is polynomial in this lower bound)

Compile a UA with weak PoK into ***new UA*** with stronger property

Note: we might get around these issues using superpolynomial simulation and/or non-standard assumptions, but we want to avoid those routes

UAoK: Weak proof of knowledge

Why is it weak?

- E produces a circuit describing the witness

where w_i is the i-th bit of the witness

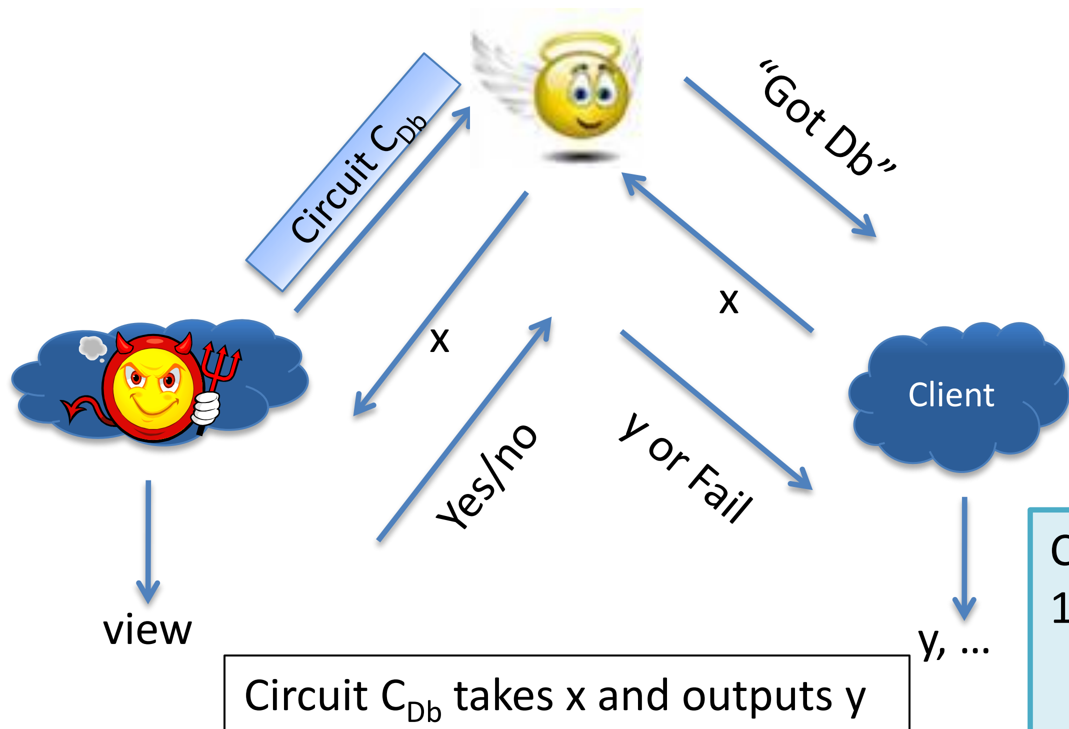
Address with modification to functionality

- If A produces a good proof with probability 1/p, E produces a good circuit with probability 1/p'
- We can't tell when C is a good circuit
  - (extracting t bits may take too long)
- E needs to be given a lower bound on the success probability of A
  - (running time is polynomial in this lower bound)

Compile a UA with weak PoK into ***new UA*** with stronger property

Note: we might get around these issues using superpolynomial simulation and/or non-standard assumptions, but we want to avoid those routes

Secure Database Commitments: Ideal model (final version)



Caveats:

1) We will require honest server to know explicit set
2) We will allow ideal parties to run in expected polytime

- Note that this is does not reduce functionality
  - Adversary is still committed to a set
  - Adversary is still required to reply consistently
  - Any polynomial sized set can be converted into a polynomial sized circuit

UAoK: Weak proof of knowledge

Why is it weak?

- E produces a circuit describing the witness

where w_i is the i-th bit of the witness

- If A produces a good proof with probability 1/p, E produces a good circuit with probability 1/p’
- We can’t tell when C is a good circuit
  - (extracting t bits may take too long)
- E needs to be given a lower bound on the success probability of A
  - (running time is polynomial in this lower bound)

Address with modification to functionality

Compile a UA with weak PoK into *new UA* with stronger property

A new tool: Universal Argument of Quasi Knowledge

- There exists extractor E

- Suppose Adv convinces verifier with probability 1/p
  1) E runs in time p * poly(k)
  2) With all but negligible probability, C is *good enough*
- Good enough: there exists valid witness $w=w_1, \dots w_t$
  – In any application, C will always* produce bits of w
  – Negligible probability that any poly-time process can find i such that

where $\omega_i \neq w_i$

Compiler for achieving quasi-knowledge

- Build UAQK from any universal argument with (slightly stronger) weak proof of knowledge property
- Gives constant round, WI UAQK based on CRHFs

This stronger property is satisfied by BG02 UAQK construction

Note: To get UAQK that succeeds with probability p, just run Adv first, and then continue with extraction iff Adv produces an accepting proof

Using UAQKs

- The idea: commit using size-hiding commitment, give a UAQK proof of knowledge of the opening
- Issues
  - UAQK extract circuit that produces bits of witness
    - But ideal input C_{Db} takes x and outputs y
  - Need contradiction if responses are not consistent with extracted database

Solution based on
- Careful formatting of witnesses
- Property-based size-hiding commitments with special structure

  - Also need a couple other pieces: statistically hiding ZKAoK, trapdoor commitments, CRHFs

Summary

Size hiding is possible in the real/ideal model.

Specifically, we can achieve secure size hiding commitments

We give:

- Definition for size hiding database commitment
- Construction which is
  - Constant round
  - Based on CRHFs
  - Non-interactive responses
- New tool: Universal Argument of Quasi Knowledge

Questions

?